CORTEX USERS GROUP

T Gray,1 Larkspur Drive,Featherstone,Wolverhampton,West Midland WV10 7TN.
E Serwa,93 Long Knowle Lane,Wednesfield,Wolverhampton,West Midland WV11 1JG.
Tel No: T Gray 0902 729078, E. Serwa 0902 732659

CORTEX USER GROUP NEWSLETTER (JULY 1987)

Issue Number 11

CONTENTS

Seaside Cottage,
Scarfskerry,
Thurso,
Caithness,
KW14 8XN.

05/05/87

Dear Tim,

Find enclosed a listing and notes for the user group. I have a few questions below, some of which are probably best answered by yourself.

Firstly, does the User Group sell the RGB interface kit for the Cortex, and if so how much is it? If not do you know where I could get one? I have recently bought a TV with an RGB facility which will save a lot of time swapping around aerial leads. On a similar note, modern TV's do not seem to have Horizontal & Vertical gain controls and I now have the problem of the screen shifting to the left which seems to be quite common. Has anyone found a solution to this problem? The suggestion on p 6.3 in Newsletter 6 did not have any noticeable effect whatsoever.

In the last newsletter you had a feature about a DIR command. I assume that the mod you mention to be able to use it is that shown on p3 of Newsletter 1. I have recently performed this modification and it seems to work. However, I have noticed that sometimes when I use DIR it works and at others it doesn't. The problem is that sometimes the leftmost nibble of some memory loacations is overwritten by a hex F. Location FEFA is where this happens most... any ideas on this?

If the memory mod has been performed is memory expansion still accessible or would the mod have to be reversed? In a similar vein would the mods to enable the centronics port to operate have to be reversed if using a 2001 replacement and the 74LS612? Also you had a RAMDISK program in the last newsletter which must use extra hardware, is any of this hardware available?

On p16 of Newsletter 4 was a list of add-ons, are any of these available? I would be VERY interested to know if anything has ever come of the M112 sound generator with keyboard mentioned by M.Rudnicki in Newsletter 2 as I also play keyboards. I am also interested in the monitor/controller in newsletter 8 so that I can use the Cortex as a sequencer for my Maplin Synthesizer. I presently use the Centronics port for this but can only send data out and not receive signals from the synthesizer. It also means I can't use the printer at the same time.

Whilst on the subject of printers, I have a Centronics 739 Printer which will not respond to your program PAINT or the DUMP program in newsletter 3. I suspect that this is because the control codes that the printer requires are different to those in your program. I enclose a copy of the codes required for graphics with the Centronics printer. Could you possibly point out as to where I could modify PAINT to make it work. Or better still, as many other people probably have similar printers a note in the next newsletter would benefit many users.

For the last week I have been playing around with your ANIMATION routine (Newsletter 2). The routine moves sprites around the screen nicely. However, if there is printing on the screen or anything plotted then some screen locations are overwritten by blank spaces. Were you aware of this?

A final note about the Newsletter. Is the section on machine code programming going to be continued. I found the previous articles very useful even though they only confirmed what I had deduced from the Cortex manual. I was hoping for further issues to explain the baffling (to me at least) practise of moving a register's contents back to itself and using some informaation thus extracted to perform conditional jumps. Example - MOVB R0,R0 followed by JEQ >FF2C as in your DIR program on p 10 newsletter 9. Such techniques cannot be deduced from the Cortex Manual and seem a bit like black magic to none experienced programmers of machine code!

Yours sincerely,

W.D.Eaves.

CORTEX USER GROUP NEWSLETTER NUMBER 11 JULY 1987

LETTERS

Bill Evans wrote such a comprehensive letter I decided to re-print it in full.

The R.G.B. interface is available from the Group as advertised in issue 10 price £8.00 for the P.C.B. or £20.00 for the full kit. The problem with the screen appearing to be two far to the left cannot be solved by modifying the Cortex as it is an inherant problem the design of the TMS9929 V.D.P. It would actually be easier to modify the T.V. set for reduced scan.

I have no idea what the problem with DIR is maybe some other user could write in and say if it works for them or not.

When the high memory mod, on page 3 newsletter 1, is done memory expansion still works normally. However part of the Centronics mod has to be removed when adding the 2001 replacement circuit. The connection between pins 18 and 19 of the 2001 socket must be removed. Two possible board are available for adding extra RAM, Ted serwa's 64K ROM-RAM card or Tim Gray's 512K DRAM card. A full list of all add-ons available at the moment was published in issue 10 we will inform everyone when more are added.

I am looking into the problem of your printer codes and hope to produce an update article for the graphics screen dump programmes in a future issue. The worst problem with your Centronics 739 printer is the limmit of 594 bytes per line as both PAINT and DUMP use 768 bytes I will have to find an alternative way of presenting the data.

We are still hoping to receive some more articles from Kevin Holloway about machine code programming. To explain the point you raise when any move or other operation is performed via the accumulator the contents are compares with zero and status bits set accordingly. So MOVB R0,R0 will move the most significant byte of R0 into the accumulator and back out again comparing it with zero as it does. Therefore the JEQ opcode will cause a jump only if the MSB of R0 was zero. JGT or JLT could have been used to determine if the value was greater than zero, a positive number, or less than zero, a negative number.

An exellent book for learning about machine code programming is "The Microcomputer Board Manual" from R.S.Components. The book is 550 pages long and describes the hardware available in the 9900 range as well as a very well written tutorial on machine code programming that can be adapted very easily. for the Cortex. The book can be purchased through the Group for £24.00 inclusive.

P Lloyd Bangor Nr Ireland

I have a 16/8 bit EPROM emulator for the E.Bus containing 16K of memory. This exists as a tag board construction, If anyone is interested I will send them a copy of the circuit in exchange for a P.C.B. design for the project.

We think you are very mean. Why not just send your design in to the User Group for publication.

REMEMBER TO SEND IN YOUR ARTICLES FOR THE NEXT NEWSLETTER

CORTEX USER GROUP NEWSLETTER NUMBER 11 JULY 1987

LETTERS

Albert Reilly Galway Eire

There are lots of different "mods" for the unobtainable 2001 can you list the advantages and disadvantages of each circuit. I would like to know more about the best expansion options, ie single or double density disks and what is the best printer.

Tim has tried to cover how the 2001 circuit and the E.Bus works in his E.Bus articles a later one will go into more detail of construction and explain some advantages. Most of the simple alternatives up to now have only provided CRU expansion and not memory expansion. The circuit published in issue 8 offers both.

The TMS9909 disc controller circuit has had problems working in double density so it is best to stick with single density for reliable operation and compitability with other users. The data separator circuit published in issue 10 solves that problem as would the new alternative disk controller board. 5" 40 and 80 track single or double density are all in use by some members but we prefer to deal in single sided single density 40 or 80 track for software exchange.

It is impossible to recommend the best printer as so many are compatible it really does depend on your own requirements.

Keith LLanegan Republic of south africa

I would like to add E.Bus and disk drives to my Cortex but am not sure if my version of the main board is suitable, also what is the best disc controller and where do I order from.

All versions of the main board are suitable for expanding for E.Bus or disk drives, however the 74LS2001 and TMS9909/9911 are no longer available. We have already printed details of a 2001 replacement in issue 8 and have a floppy controller card that replaces the TMS9909, TMS9911 and all the other disk interface components. It uses the socket vacated by the TMS9909 to connect to the mother board. The card is available for £40.00 from the group.

\- - - - - - - - - - - - - -

EDITORIAL

The group is going through a very busy time just lately with letters arriving every day. We hope to answer all of them eventually but due to the limited space in the newsletter it may not be as soon as you would prefer. If you enclose a stamped adressed envelope with your letter a reply will be sent within a week or so. All orders for components or kits are best sent direct to Ted Serwa who deals with this side of things.

The user group will be holding another combined meeting with the TI994A user group in Bloxwich near wolvrhampton ON SATURDAY September 5th. We will send more details in the next issue.

REMEMBER TO SEND IN YOUR ARTICLES FOR THE NEXT NEWSLETTER

SOLVING SYSTEMS OF LINEAR EQUATIONS

The following program gives an efficient method for solving systems of linear equations. In Newsletter number 7 K.P. Holloway described one algorithm for doing this. However, his algorithm is unduly complicated and, even worse, is quite inefficient. This algorithm is some 2.5 to 3 times faster than his.

Furthermore, this algorithm is split into two parts so that solving $Ax = b$ for more than one right-hand-side b is extremely efficient. The first part of the routine starts at line 100 and calculates an LU factoring of the matrix A (with partial pivoting), this part of the routine must be called whenever the matrix A is changed (but need not be called apart from that). The second part of the routine starts at line 300 and calculates the solution vector x of $Ax = b$ for a given vector b. It can be called repeatedly with many different vectors b.

The communication between the two parts is done through the matrix A which holds the LU factorisation of A and a permutation vector P which describes the pivoting that was done. These must not be changed in-between the two parts.

The code starting at line 500 just sets up an example matrix A and vector b.

```
 10 GOSUB 500
 20 GOSUB 100
 30 GOSUB 300
 40 FOR I=1 TO N: ? "X[";I;"] =",X[I]: NEXT I
 50 STOP
100 REM Do LU factoring of A using P
120 DIM P[N]: FOR I=1 TO N: P[I]=I: NEXT I
130 FOR K=1 TO N-1
140 REM pick largest element to pivot on
150  J=K
160  FOR I=K+1 TO N
170   IF ABS[A[P[I],K]] > ABS[A[P[J],K]] THEN J=I
180  NEXT I
190  I=P[J]: P[J]=P[K]: P[K]=I
200  REM pivot
210  FOR I=K+1 TO N
220   PV=A[P[I],K]/A[P[K],K]
230   A[P[I],K]=PV
240   FOR J=K+1 TO N
250    A[P[I],J]=A[P[I],J]-PV*A[P[K],J]
260   NEXT J
270  NEXT I
280 NEXT K
290 RETURN
```

```
300  REM Use LU factoring to solve Ax=B
310  REM returning result in X, uses P.
315  DIM X[N]
320  REM forward reduction
330  FOR K=1 TO N-1
340   FOR I=K+1 TO N
350    B[P[I]]=B[P[I]]-A[P[I],K]*B[P[K]]
360   NEXT I
370  NEXT K
380  REM backward substitution
390  FOR K=N TO 1 STEP -1
395   X[K]=B[P[K]]
400   FOR I=K+1 TO N
410    X[K]=X[K]-A[P[K],I]*X[I]
420   NEXT I
430   X[K]=X[K]/A[P[K],K]
440  NEXT K
450  RETURN
500  REM SET UP ARRAYS
510  RESTOR 610
520  READ N
530  DIM A[N,N],B[N]
540  FOR I=1 TO N
550   FOR J=1 TO N
560    READ A[I,J]
570   NEXT J
580   READ B[I]
590  NEXT I
600  RETURN
610  DATA 4
640  DATA 0,1,3,0,7
650  DATA 1,1,1,1,6
660  DATA 1,0,2,0,4
670  DATA 1,1,0,1,4
 RUN
X[ 1] =  0
X[ 2] =  1
X[ 3] =  2
X[ 4] =  3
 Stop at 50
```

DATA FILES by John Mackenzie

SEQUENTIAL RELATIVE DATA FILES

1. A sequential file places data into records one after the other on the disk. The record length is determined by the amount of data you put in it.

2. To Create a sequential file:

```
OPEN D,"NAME",F,CRE
```

D - drive number. "NAME" or $ variable. F - file variable.

3. Once a file exists on a disk all you do to open it is:

```
OPEN D,"NAME",F
```

4. Data is then stored in the records by putting variables into them with:

```
PUT F,$B,$C,D,- etc      or      PUT F,$B
                                 PUT F,$C
                                 etc
```

F - file variable from the OPEN statement.

a. A numeric or $ variable can be saved in this way.
b. The maximum number of characters that will be passed by a numeric variable is 6, even if the variable has been dimensioned larger. To pass more than 6 characters use a $ variable.
c. Note in sequential files the $ sign remains in the PUT statement syntax.
d. In the second option above the PUT statement can be seperated by other program lines.

5. When all the variables have been put to the disk then:

```
CLOSE F
```

This is critical in sequential files to maintain the directory and so that when you open the file the next time, the file control mechanism is reset.

6. Data is retrieved from the records by:

```
OPEN D,"NAME",F
GET F,$B,$C,$D,- etc     or      GET F,$B
                                 GET F,$C
                                 etc

CLOSE F
```

Again the close is critical.

7. Example: `DIM $A[13,14]`

To save.

```
OPEN 1,"FILE",F
FOR X = 0 TO 13
  PUT F,$A(X,0)
NEXT X
CLOSE F
```

To retrieve.

```
OPEN 1,"FILE",F
FOR X = 0 TO 13
  GET F, $A(X,0)
NEXT X
CLOSE F
```

8. If you think of a sequential file as a long snake of variables placed end to end, you can see why it is impossible to select one in the middle, see summary.

9. Sequential files make the best use of your disk space.

RANDOM RELATIVE DATA FILES

1. A random file places data into records of a predefined size in the order you require.

2. To create a random file, first decide the maximum number of characters that any one of the variables to be saved will hold. As an example I will assume 50.

```
OPEN D,"NAME",F,CRE,50
```

The only addition to the sequential create is the record size. It is this during the create statement that causes the file control mechanism to create a random file.

3. To open a file that exists on a disk:

```
OPEN D,"NAME",F
```

4. Data is stored in the records with:

```
PUT F,R,B,C,D,- etc
```

or

```
DIM $A(13,14)
FOR X = 0 TO 13
  PUT F,X,A(X,0)
NEXT X
```

R - variable containing the record number; or the number.
B,C,D,- etc - variables to be saved.

a. The data in the variable can be any length up to the record size.
b. If you put to record 500 without using the lower order records space on the disk is allocated for those lower order records.
c. The syntax for the random files requires the $ sign to be dropped.
d. Numeric variables with more than 6 characters can be stored in random files.
e. In the first PUT option above the record R is used for the first variable B, the variables C,D,- etc are then stored in successive records.

5. When all the data has been passed then:

```
CLOSE F
```

The close statement is only essential after the first time the file has been used, or if more records have been used than the last time the file was closed.

6. To retrieve data just replace the PUT's above with GET's. the rules are the same. (drop the $ sign)

7. Should you attempted a get from a record you have not put to, you will get a READ OUT OF DATA error. This is important in that should you put to a high order record the lower order records are not initialised. To initialise you only have to put a $ variable that contains "" (nothing), or a similar numeric variable.

8. If you think of a random file as a stack of records placed one on top of the other, it is easy to select any record, see summary.

9. Random files are greedy on disk space.

SUMMARY

	SEQUENTIAL:	RANDOM
Create:		
	`OPEN D,$A(0),F,CRE`	`OPEN D,$A(0),F,CRE,80`
Open:		
	`OPEN D,$A(0),F`	`OPEN D,$A(0),F`
Save:		

	`PUT F,$B,C,$D, etc`
	or
	`FOR X = 0 to Y`
	`  PUT F,$B(X,0)`
	`NEXT X`
		`PUT F,5,B(5,0),B(6,0), etc`
		or
		`FOR X = 0 TO Y`
		`  PUT F,X,B(X,0)`
		`NEXT X`

Close:		
	`CLOSE F`	`CLOSE F`
Retrieve:		
	Replace PUT's above with GET.	
Notes:		
	Always CLOSE! Use $ sign. PUT & GET all data. For more than 6 char's use $ variables.	CLOSE after first use. CLOSE after more records used. Drop $ sign. PUT & GET only record required.

PICTORIAL VIEW OF FILES

To save: V1- 6 Chars, V2- 8 Ch', V3- 4 Ch', V4- 10 Ch', V5- 6 Ch'.

Sequential File:

xxV1xx	T	xxxV2xx	T	xV3x	T	xxxxV4xxxx	T	xxV5xx

T - Terminator

Random File:

Record No	0	xxV1xx
"	1	xxxV2xxx
"	2	xV3x
"	3	xxxxV4xxxx
"	4	xxV5xx

Largest variable = 10 char's
Record size on create = 10

INTO ONTO AND OUT OF E.Bus Part 2. T.Gray

As mentioned in my first article I will try to explain more fully what happens at the memory card end of the E.Bus. The card discussed is a modified version of Ted Serwas 64K ROM card that now has selectable wait states and the capability of being used with 2764 EPROM or 6264 SRAM mixed in any combination.

When an external memory access is requested the the E.Bus lines A0 to A15 will contain the memory address, XA0 to XA3 will contain the extended address and -memen will be low signifying a memory request. IC1 compares the extended address lines with the code set up on S1 and produces a BOARD ENABLE signal when the extended address is relevant to this board. S1 allows the board to be located anywhere within the 1MB E.Bus memory address range

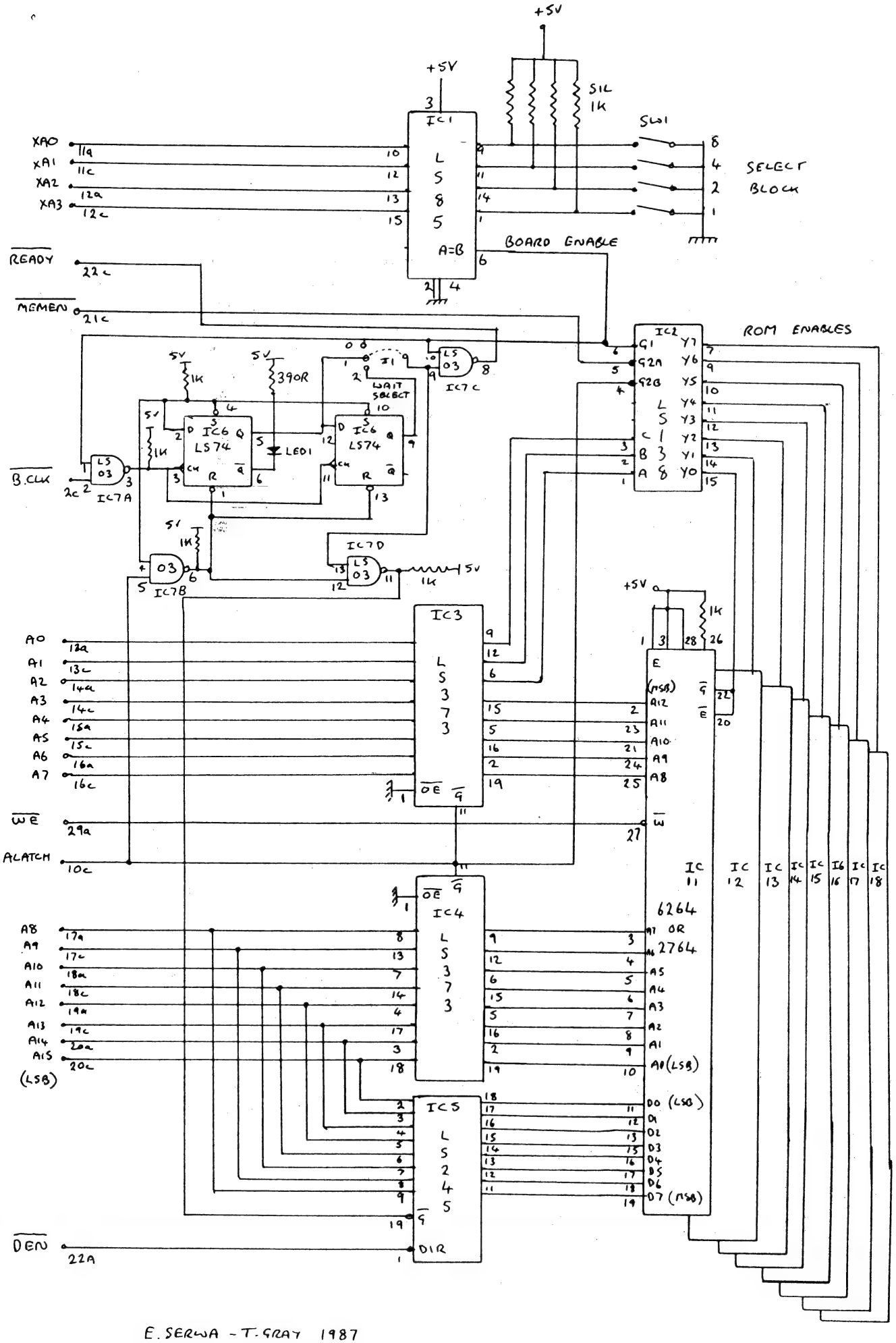
The normal address is passed through the LS373 transparent latch ICs with the top three bits connected to chip select logic IC2. When ALATCH goes low on the next clock cycle one of the outputs of IC2 (depending on the address) will go low selecting one of the memory ICs. At the same time either -WE or -DEN will go low depending on whether the memory request is a write or a read. The -DEN signal is used to control the direction of the data buffer IC5.

As most memory ICs are too slow to present data within one clock cycle the ready signal is generated one or two clock cycles after ALATCH goes low by the circuitry arround IC6. Jumper J1 selects one or two wait states. The output from IC6 passes through IC7c to form the -B.READY signal for the Cortex and at the same time through IC7d to enable the data buffer. If the memory access were a write access -DEN would stay high and the data from the Cortex would pass through IC5 to the memory ICs, at the same time -WE would go low writing the data into the RAM.

The board will actually work with IC5 wired out and the memory ICs data lines connected directly to the bus, but as the E.Bus design handbook dictates that a maximum of two input or tri-state output lines should be connected to the bus lines, on any one board, it was thought best to include it. For the same reason the -WE signal should realy be buffered before going to the memory ICs.

A P.C.B. is available for this project from Ted Serwa C/O the user group and I have also designed a 512K DRAM card using a TMS4500 and sixteen 41464 DRAM ICs please write to me, Tim Gray if you would be interested in a P.C.B.

64K ROM/RAM E.Bus Expansion Card using 2764 ROM's or 6264 RAM's



E. SERWA - T. GRAY 1987

"STAR" EPROM T. GRAY 1987

I have written some STAR commands for Cortex basic designed to be used from an EPROM on the E.Bus system. Most of the routines are based on articals published in CORTEX USER GROUP magazines and are modified for use as a STAR routine. The EPROM is available for £5.00 from Tim Gray. The screen dump routines may need to have the printer set codes altering for your printer please supply details of codes for setting bit image printing modes and setting line spacing and margin as well as the unit number that you use for printer. The *DIR command expects drive 3 to be RAMDISK on the E.Bus please state if this is required and if so what address the RAM starts at.

The routines are as follows.

*FILL — Fills or unfills an enclosed area of the screen until a different colour is reached.
*FILL <start X>,<start Y>

*COPY — Makes one graphics cell an exact copy of another by changing the VDP name table the copy cell will always be the same as the master even if the master is later changed.
*COPY <copy cell>,<master cell>
Note they must be in the same vertical third of the screen.

*COLOUR — Allows setting all the lines of a cell colour individually.
*COLOUR <cell>,<col>,<col>,<col>,<col>
Where col is the Hex code for fg/bg colour of two lines at a time. Ie 7E1AH sets colour of first line to 7,14 and second line to 1,10.

*DIR — Lists the directory of a disk without corupting the Basic programme.
*DIR <drive number>

*VAR — Lists all the variable names used in the Basic programme in memmory all dimentiones are shown if the variable has been dimentioned.
No parameters.

*PAINT — Sends graphics screen to the printer as a bit image picture with colours shown as diferent grey levels.
No parameters.

*PAINTS — As above but sideways on the page to fill an A4 sheet of paper.
No parameters.

*DUMP — Sends graphics screen to printer as a direct bit image with no grey scale.
No parameters.

*SCROLL

Scolls the graphics screen left by one pixel.
*SCROLL <start line>,<end line>

*LCASE

Loads a true lower case character set.
No parameters.

*SCREEN

Laods/saves the whole 16K of screen memmory
to or from main memmory.
DIM SCN[2730] to reserve space.
*SCREEN 0,ADR[SCN[0]] to save
*SCREEN 1,ADR[SCN[0]] to load

*SWAP

A better way to use the SWAP routine.
*SWAP <old colour>,<new colour>
changes old colour for new colour over the
whole screen.

*POLYGON

Draws a polygon of N number of sides between
3 and 50.
*POLYGON <XC>,<YC>,<XO>,<YO>,<N>,<SF>,<MODE>
XC,YC centre of polygon.
XO,YO one of the outside points.
N number of sides.
SF aspect ratio 10 to 1000 normal is 180
MODE 0 normal plot
1 unplot
2 overplot (forground colour only)
3 XOR plot (no colour change)

*CIRCLE

As above but gives a better circle.
*CIRCLE <XC>,<YC>,<XO>,<YO>,<0>,<SF>,<MODE>

*ARC

Draws part of a circle to form an arc.
*ARC <XC>,<YC>,<XO>,<YO>,<DEG>,<SF>,<MODE>
DEG number of degrees 1 to 360

*LINE

Draws a line using the plotting mode options.
*LINE <X1>,<Y1>,<X2>,<Y2>,<MODE>

*BOX

Draws a rectangular shape with plotting mode
and fill/normal options.
*BOX <X1>,<Y1>,<X2>,<Y2>,<MODE>,<FILL>
FILL 0=open 1=solid

*FIND

Requests the sting to be found and lists all
occurances of the string in the Basic programme.

CORTEX HARDWARE MODIFICATIONS

W.R. BUCKNALL
HAMPSHIRE

DISK INTERFACE

1. Replace IC70, 74LS123, with a 74LS221 to reduce the possibility of false triggering. The 221 is a one-shot & therefore cannot be re-triggered by noise, also it has Schmitt i/p giving high noise immunity. To do this the timing capacitor must be connected between pins 6 & 7 (not 7 & ground) & the link between pin 6 & ground must be cut. Replace R69 & R70 with trimpots if you can. I found that pulse widths of 1.54µs & 0.6µs give error free operation for SD & DD respectively. If you have no intension of using 8" disks then R68 & D2 may be removed. In the TTL data book for the 74LS163 it says, "The use of the ripple carry output as an edge trigger is not recommended." To overcome this two Schmitt inverters may be connected between pin 15 IC 87 & pin 11 IC69. A 74LS14 can be piggy-back mounted on IC69, to sharpen the edge & introduce a delay to allow the 74LS163 to settle.

2. If you have Canon 80 track drives & wish to use 40/80 track switching then you'll find the square $\overline{\text{STEP}}$ o/p of the TMS9909 is not suitable. To overcome this problem piggy-back mount a 74121 & timing components (approx 1ms) on IC83, cut the $\overline{\text{STEP}}$ track between the FDC & IC83. The $\overline{\text{STEP}}$ o/p from the TMS9909 is used to trigger the 74121 with its o/p being fed to the drive through the 7407.

   More details of this & of Canon's MDD220, 210, & 110 may be obtained from the above address.

CASSETTE INTERFACE.

1. As IC70 is now changed then the cassette interface has to be modified. The cassette interface will have increased stability due to the same reason described for the disk interface above, to give few read errors.

2. Cut the link between pin 14, IC70, & ground & change C20 for a 47n & R44 for a 10K to give a pw of 329µs (nearly 312µs). Remove C32 (1n between pins 1 & 2 IC70) & decrease C21 (all pass filter) from 4n7 to 0n5 to improve the rise time.

3. Insert a resistor (SOT) between pin 2 IC71, 75189, & ground. This increases the threshold of the level shifter, hence reduce errors caused by noise. Its value will depend on your cassette player o/p (for my ghetto blaster I fitted 1k) Decouple IC72 (0µ1 across pins 4 & 11).

4. Due to the high impedance of the interface a load resistor must be fitted across the tape i/p to match it to the low impedance of the cassette player. The value will depend on your player. If a long lead is used, to reduce the noise induced in the lead, R47 may be increased from 100R to 1k & a 100R fitted at the cassette end of the lead.

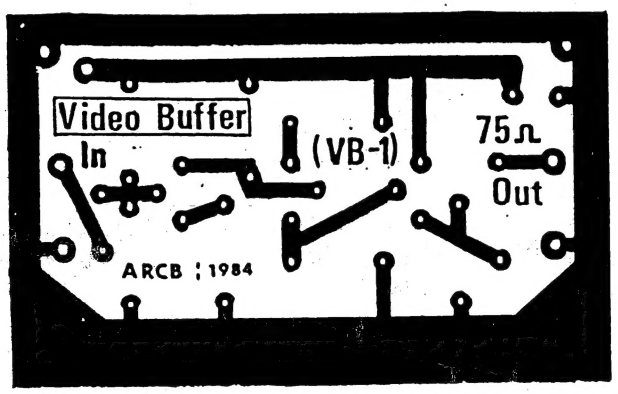
VDP.

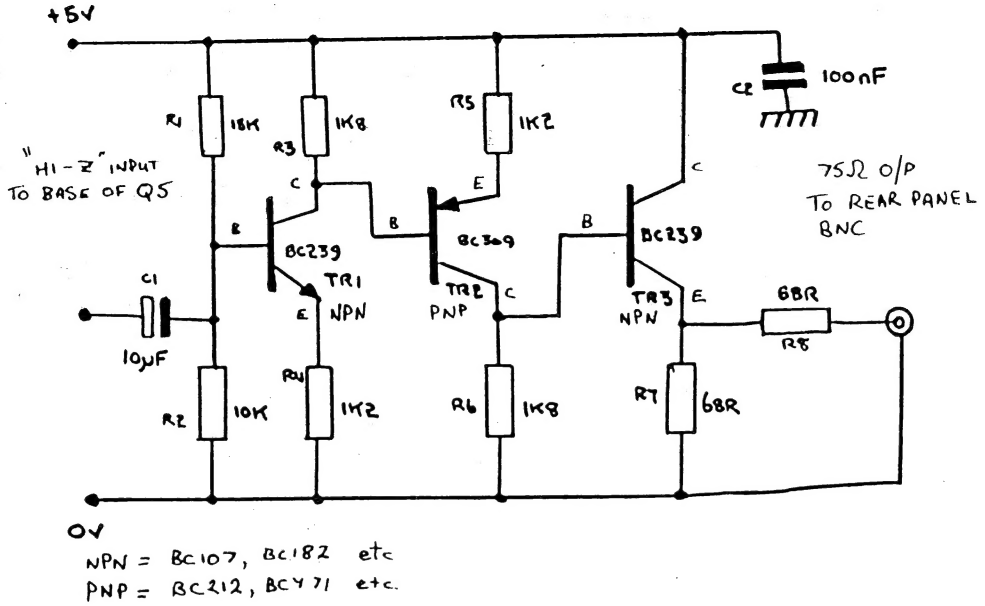
1. The VDP has no decoupling, a 100n may be fitted underside between pins 12 & 33. The TMS9929 recommends the o/p's be loaded with 560R (470R TMS9919), therefore if you have fitted the RGB i/f then remove either R20/21, R16, & R17 on the main board or R16, R17, & R47 on RGB board. Tim Gray's modification may be implemented using IC79c as the sync sep.

Alen Badcock has sent in this circuit for a composite video output buffer.

Video monitors with composite video inputs are more common these days and this buffer allows the cortex to be used with one thereby bypassing the R.F. Modulator and T.V. Tuner prosses.

CORTEX COMPOSITE VIDEO BUFFER





NPN = BC107, BC182 etc
PNP = BC212, BCY71 etc.

CORTEX USERS GROUP SOFTWARE

ALL GAMES £2.50 EACH

BURGLAR	MUNCHER
FROGGER	G DESIGN
INVADERS&ASTEROIDS	WALL
HUNCHBACK	ARCHIE
MAZE	NIGHT ATTACK
OLIMPICS NEW	WINE&FORM1
FIRE BIRD	SHARES&HISTOGRAM
PENGO	MOONBASE II
LABYRINTH OF TRAG	CENTIPEDE

MEMBERS ADVERTS

MENU GENERATOR - VERSION 1.0

A SUIT OF PROGRAMS TO DESIGN A CUSTOM 'MENU' ENVIROMENT FOR A TARGET APPLICATIONS DISK . THIS WILL ALLOW APPLICATIONS PROGRAMS , GAMES OR UTILITIES TO BE USED BY WAY OF A SIMPLE SELECTION METHOD . THE TARGET MENU GIVES NEAT PRESENTATION OF YOUR PROGRAMS WITH A DISPLAY OF TITLES AND COPYRIGHT . THE DEVELOPMENT DISK CONTAINS ALL THE TOOLS AND OPERATING FILES TO GENERATE A CUSTOM MENU , AND USE A SIMIILAR MENU STRUCTURE ITSELF

AVAILABLE IN 40 TRACK , SINGLE OR DOUBLE DENSITY,SINGLE OR DOUBLE SIDED

PRICE £10 ALL INC FROM---A.R.C BADCOCK
'WINDSONG'
7 HEATHFIELD ROAD
HILTINGBURY
CHANDERS FORD
HANTS SO5 1RP